Design Configuration Subsystems Correctly and Distribute Safe Default Configurations

William L. Fithen, Software Engineering Institute [vita[3]]

2005-10-03

Poorly designed configuration subsystems and poor default configurations may produce system vulnerabilities.

Description

The configuration of a system is the non-executable data delivered with a system that governs its dynamic behavior. The configuration is generally a set of variable values that supply information to the system in order to customize its behavior for a particular environment.

Default Configurations

This occurs when a system is shipped with a default configuration <define> <j2ee XML aspect oriented programming> that is insecure [Schneier 02: II. Default Configurations].

While this is not a programming vulnerability, it is an engineering vulnerability that is introduced in the product packaging phase of the software development life cycle.

Management or Debugging Interfaces Left Enabled

Administrative interfaces can lead to vulnerability. Sometimes the interface is left in by mistake. Sometimes it is intentional, but insecure.

In many cases, such administrative interfaces are configurable. In general, the solution to such interfaces is to configure them off. If, however, the product ships with such interfaces enabled by default, then one would reasonably classify this as a vulnerability in the product (if not in the software per se).

Administrative or management interfaces should always be restricted (via [authentication[12], authorization[13]]) to proper administrators or managers.

Configuration Languages Too Complex

When the configuration "language" of a system is too complex, insufficiently expressive, contradictory, misleading, or ambiguous, it is reasonable to argue that this design will produce deployed systems that are vulnerable.

For example, avoid double or triple negatives, such as

```
no-read: false
```

[3] daisy:320 (Fithen, William L.)

[12] daisy:321 (Use Authentication Mechanisms, Where Appropriate, Correctly)

[13] daisy:322 (Use Authorization Mechanisms Correctly)

When complex configuration languages are necessary,[19] be sure to include in system adequate tooling for creating, managing, and checking such configuration files.

References

[Landwehr 93] Landwehr, Carl; Bull, Alan; & McDermott, John. "A Taxonomy of Computer Program Security Flaws, with Examples." Technical report NRL/FR/5542--93/9591. United States Navy, Naval Research Laboratory, Nov. 1993.

[Schneier 02] Schneier, Bruce. "Judging Microsoft." *Crypto-Gram Newsletter*. February 15, 2002. http://www.schneier.com/crypto-gram-0202.html

[VU#247371] *Vulnerability Note VU#247371: Borland/Inprise Interbase SQL database server contains backdoor superuser account with known password*. cert.org, 2001. http://www.kb.cert.org/vuls/id/247371.

[VU#602734] *Vulnerability Note VU#602734: Cisco default install of IBM Director agent fails to authenticate users for remote administration*. cert.org, 2004. http://www.kb.cert.org/vuls/id/602734.

[VU#858726] *Vulnerability Note VU#858726: MailPost discloses sensitive system information when operating in debug mode*. cert.org, 2004. http://www.kb.cert.org/vuls/id/858726.

[1[25]]

SEI Copyright

Felder

Name	Wert

[19] For example, J2EE deployment descriptors or Java Aspect Oriented Programming directives.

[25] file:///Users/wlf/Workspaces/Eclipse-3.1/swa-content/documents/html-upload/knowledge/guidelines/configuration.html#d0e119

[1] http://www.sei.cmu.edu/about/legal-permissions.html

Copyright Holder	SEI

Felder

Name	Wert
is-content-area-overview	false
Content Areas	Knowledge/Guidelines
SDLC Relevance	Implementation
Workflow State	Publishable